

# Informe Técnico: Análisis del Flujo de Datos y Protocolos Activos en una comunicación web

El presente informe detalla el análisis del flujo de datos y la interacción de protocolos clave en una comunicación web simulada, utilizando el modelo TCP/IP como marco de referencia. El objetivo es comprender los roles de los diferentes protocolos desde que un usuario ingresa una URL (<http://intranet.miempresa.local>) hasta que la página web carga completamente, en un escenario de acceso a un portal interno simulado localmente.

## 1. Escenario de la Comunicación Web Simulada

Se simula el acceso a un portal interno de la empresa, <http://intranet.miempresa.local>. La tarea implica observar y analizar los protocolos que participan en cada capa del modelo TCP/IP.

El entorno de simulación se estableció localmente, utilizando Docker Compose para desplegar un servidor web Nginx que sirve un archivo `index.html` básico, y modificando el archivo `hosts` del sistema para mapear [intranet.miempresa.local](http://intranet.miempresa.local) a `127.0.0.1`.

### Configuración del entorno (según el ejercicio):

- **Archivo `docker-compose.yml`:**

YAML

```
version: '3.8'
```

```
services:
```

```
  nginx:
```

```
    image: nginx:latest
```

```
    ports:
```

```
      - "80:80" # Mapea el puerto 80 del host al puerto 80 del contenedor Nginx
```

```
    volumes:
```

```
      - ./html:/usr/share/nginx/html # Monta la carpeta 'html' local dentro del contenedor
```

- **Archivo `html/index.html`:**

HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Intranet Segura</title>
```

```
</head>
```

```
<body>
```

```

<h1>¡Bienvenido a la Intranet Segura de mi empresa!</h1>
<p>Esta es una página de prueba para demostrar el acceso seguro.</p>
</body>
</html>

```

- **Modificación del archivo hosts:** Se añadió la línea 127.0.0.1 intranet.miempresa.local para resolver localmente el nombre de dominio sin necesidad de un servidor DNS externo.

## 2. Análisis de la Ruta de Datos en Capas TCP/IP

El siguiente cuadro detalla los protocolos que intervienen en cada capa del modelo TCP/IP durante una sesión de navegación web (HTTP), y describe su función principal.

Capa TCP/IP	Protocolo(s) Usado(s)	Función Principal
<b>Aplicación</b>	HTTP, DNS	<b>HTTP:</b> Protocolo para la transferencia de hipertexto, utilizado por el navegador para solicitar y mostrar la página web. &lt;br> <b>DNS:</b> (Simulado por hosts en este caso) Resuelve nombres de dominio (ej. intranet.miempresa.local) a direcciones IP.
<b>Transporte</b>	TCP	<b>TCP (Transmission Control Protocol):</b> Proporciona una conexión fiable, orientada a la conexión, garantizando que los datos lleguen en orden, sin errores y sin pérdidas. Utiliza puertos (ej. 80 para HTTP) para identificar aplicaciones.
<b>Internet</b>	IP	<b>IP (Internet Protocol):</b> responsable del direccionamiento lógico y el enrutamiento de paquetes a través de la red, permitiendo que los datos viajen desde el origen hasta el destino a través de diferentes redes (en este caso, localmente a 127.0.0.1).
<b>Acceso a Red</b>	Loopback (Localhost)	Cuando se usa 127.0.0.1, el tráfico no sale a una interfaz física de red. En un entorno real, sería <b>Ethernet/Wi-Fi:</b> Protocolos que gestionan la transmisión física de bits y el control de acceso al medio. Aquí se encuadran las direcciones MAC y la interacción con los switches y tarjetas de red.

### 3. Captura y Observación de Paquetes

Para una comprensión más profunda, se realizó una captura de red utilizando Wireshark mientras se accedía a <http://intranet.miempresa.local>. Se observaron los siguientes paquetes clave:

**Top Screenshot: Loopback Traffic Capture**

Wireshark Adapter for loopback traffic captureKNI82.pcapng

Paquetes: 62 · Displayed: 8 (12.9%)

Perfil: Default

Filter: `ip.addr == 127.0.0.1`

No.	Time	Source	Destination	Protocol	Length	Info
46	29.610727	127.0.0.1	127.0.0.1	TCP	56	56812 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
47	29.610750	127.0.0.1	127.0.0.1	TCP	44	80 → 56812 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
48	29.611076	127.0.0.1	127.0.0.1	TCP	56	56813 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
49	29.611090	127.0.0.1	127.0.0.1	TCP	44	80 → 56813 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
50	32.692976	127.0.0.1	127.0.0.1	TCP	56	56815 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
51	32.692994	127.0.0.1	127.0.0.1	TCP	44	80 → 56815 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
52	32.693162	127.0.0.1	127.0.0.1	TCP	56	56816 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
53	32.693175	127.0.0.1	127.0.0.1	TCP	44	80 → 56816 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Frame 46: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF\_{...}, id 0

Null/Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 56812, Dst Port: 80, Seq: 0, Len: 0

**Bottom Screenshot: Wi-Fi Traffic Capture**

Wireshark\_Wi-Fi57V082.pcapng

Paquetes: 158837 · Displayed: 21 (0.0%)

Perfil: Default

Filter: `tcp.port == 80`

No.	Time	Source	Destination	Protocol	Length	Info
34377	204.627002	2803:c180:2100:6f02::...	2600:1419:200:886::...	TCP	86	56210 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
34378	204.632545	2600:1419:200:886::...	2803:c180:2100:6f02::...	TCP	86	80 → 56210 [SYN, ACK] Seq=0 Ack=1 Win=64800 Len=0 MSS=1392 SACK_PERM WS=128
34379	204.632595	2803:c180:2100:6f02::...	2600:1419:200:886::...	TCP	74	56210 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
34380	204.632696	2803:c180:2100:6f02::...	2600:1419:200:886::...	HTTP	475	GET /MFewTzBNMEswSTA3BgUrDgMGcGUABBRz2bWARTxMtEy9aspRAZg5QfhagQQUgrWPZfOn89x6JI3r%2F2ztwk1V88CE
34381	204.640691	2600:1419:200:886::...	2803:c180:2100:6f02::...	TCP	74	80 → 56210 [ACK] Seq=1 Ack=402 Win=64512 Len=0
34382	204.643300	2600:1419:200:886::...	2803:c180:2100:6f02::...	HTTP	432	HTTP/1.1 304 Not Modified
34387	204.660567	2803:c180:2100:6f02::...	2a04:4e42:34::684	TCP	86	56211 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
34388	204.667110	2a04:4e42:34::684	2803:c180:2100:6f02::...	TCP	86	80 → 56211 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1392 SACK_PERM WS=512
34389	204.667176	2803:c180:2100:6f02::...	2a04:4e42:34::684	TCP	74	56211 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
34390	204.667280	2803:c180:2100:6f02::...	2a04:4e42:34::684	HTTP	356	GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?3751854372854d05 HTTP/1.1
34391	204.676999	2a04:4e42:34::684	2803:c180:2100:6f02::...	TCP	74	80 → 56211 [ACK] Seq=1 Ack=283 Win=145408 Len=0
34392	204.676999	2a04:4e42:34::684	2803:c180:2100:6f02::...	HTTP	277	HTTP/1.1 304 Not Modified
34393	204.695274	2803:c180:2100:6f02::...	2600:1419:200:886::...	TCP	74	56210 → 80 [ACK] Seq=402 Ack=359 Win=65024 Len=0
34394	204.719674	2803:c180:2100:6f02::...	2a04:4e42:34::684	TCP	74	56211 → 80 [ACK] Seq=283 Ack=204 Win=65280 Len=0
79225	264.680044	2803:c180:2100:6f02::...	2a04:4e42:34::684	TCP	74	56211 → 80 [FIN, ACK] Seq=283 Ack=204 Win=65280 Len=0
79226	264.680119	2803:c180:2100:6f02::...	2600:1419:200:886::...	TCP	74	56210 → 80 [FIN, ACK] Seq=402 Ack=359 Win=65024 Len=0
79227	264.685346	2600:1419:200:886::...	2803:c180:2100:6f02::...	TCP	74	80 → 56210 [FIN, ACK] Seq=359 Ack=403 Win=64512 Len=0
79228	264.685371	2803:c180:2100:6f02::...	2600:1419:200:886::...	TCP	74	56210 → 80 [ACK] Seq=403 Ack=360 Win=65024 Len=0
79229	264.685917	2a04:4e42:34::684	2803:c180:2100:6f02::...	TCP	74	80 → 56211 [ACK] Seq=204 Ack=284 Win=145408 Len=0
79230	264.686239	2a04:4e42:34::684	2803:c180:2100:6f02::...	TCP	74	80 → 56211 [FIN, ACK] Seq=204 Ack=284 Win=145408 Len=0
79231	264.686258	2803:c180:2100:6f02::...	2a04:4e42:34::684	TCP	74	56211 → 80 [ACK] Seq=284 Ack=205 Win=65280 Len=0

Frame 34377: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF\_{5B393745-28C6-4227-85BA-24D82F3432FD}

Ethernet II, Src: Intel\_dd:0d:7e (3c:e9:f7:dd:0d:7e), Dst: HuaweiTechno\_f7:5e:11 (c0:e3:fb:f7:5e:11)

Internet Protocol Version 6, Src: 2803:c180:2100:6f02:d002:2248:131a:9411, Dst: 2600:1419:200:886::1b01

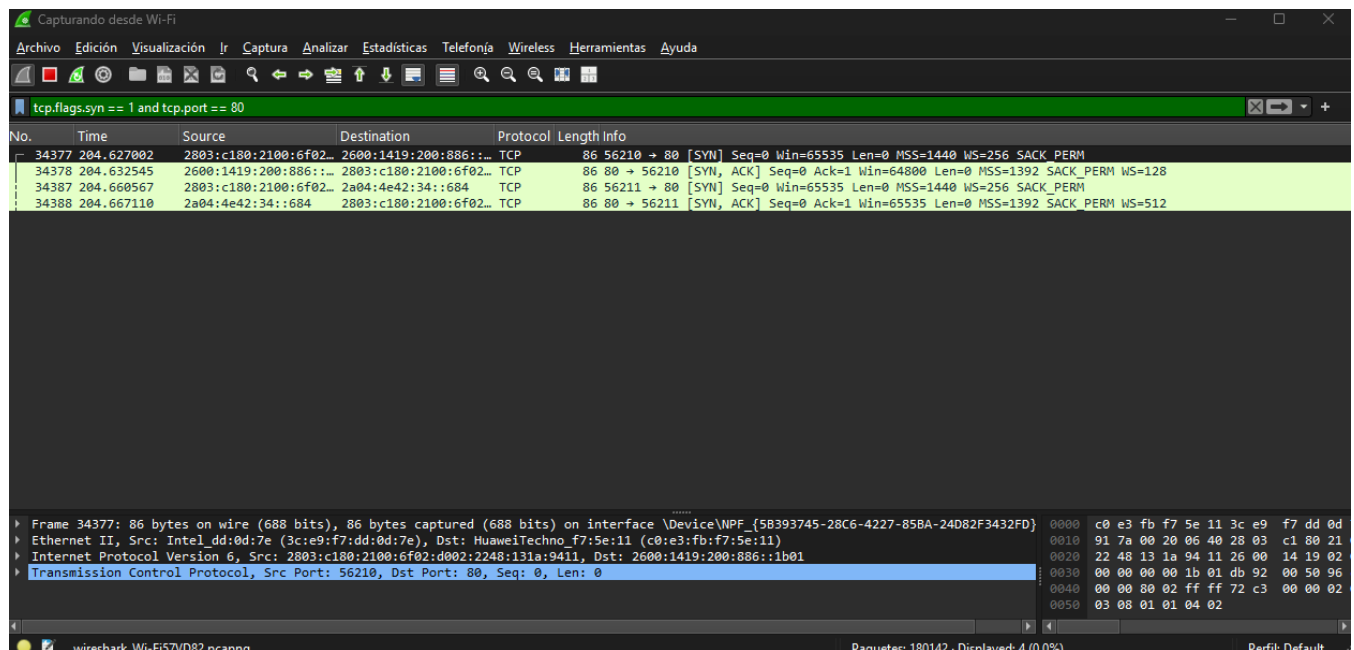
Transmission Control Protocol, Src Port: 56210, Dst Port: 80, Seq: 0, Len: 0

- **Resolución de Nombre de Dominio (vía archivo hosts):** Dado que el nombre intranet.miempresa.local se mapea directamente a 127.0.0.1 en el archivo hosts del sistema, no se genera una consulta DNS a través de la red que Wireshark pueda capturar en el puerto UDP 53. Sin embargo, la resolución es confirmada por comandos del sistema operativo.

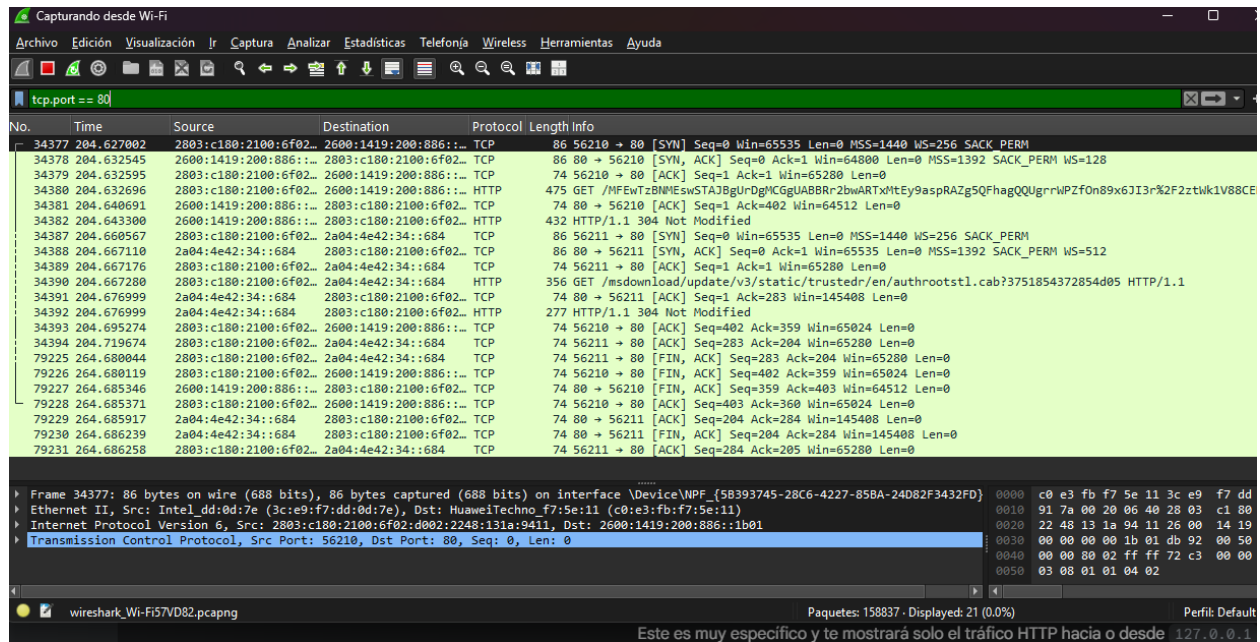
```
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

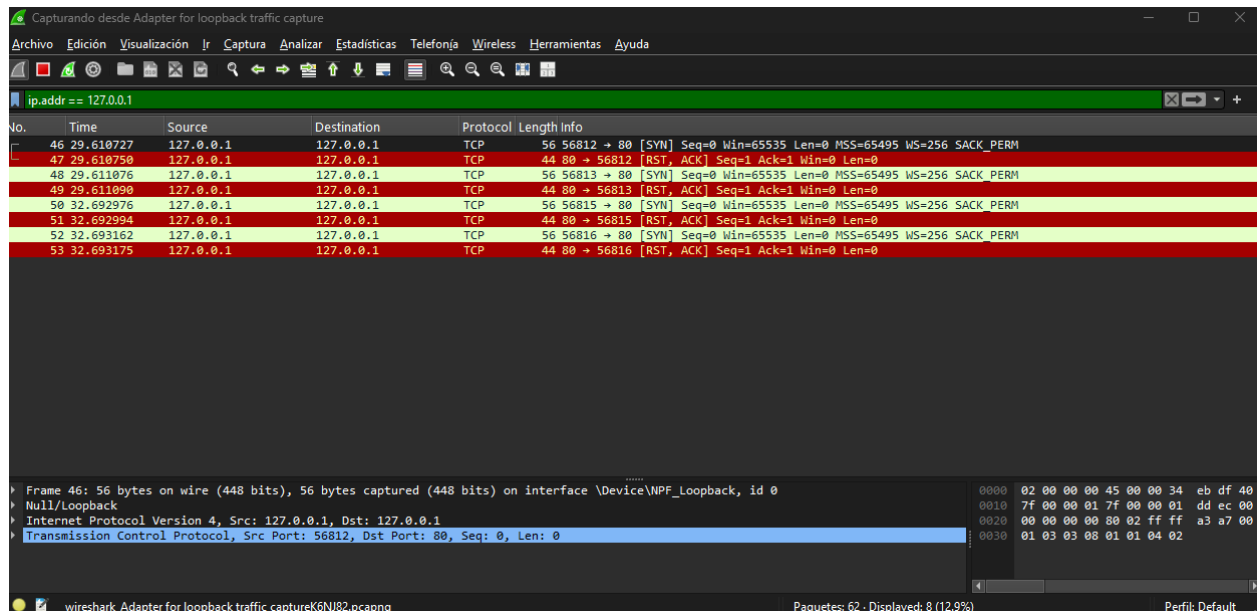
- **Paquete TCP (Establecimiento de Conexión):** Se identificó el establecimiento de una conexión TCP (handshake de tres vías - SYN, SYN-ACK, ACK) en el puerto **80** (para HTTP) con la IP 127.0.0.1, confirmando la fase inicial de la comunicación fiable.



- **Paquete HTTP (Solicitud y Respuesta):** Se visualizaron las solicitudes GET del navegador para obtener el contenido de la página y las respuestas HTTP/1.1 200 OK del servidor Nginx, conteniendo el HTML de la página.



- **Tráfico IP general:** Se observó el tráfico a la dirección IP local 127.0.0.1, confirmando que la comunicación se mantuvo dentro de la propia máquina y no salió a la red externa.



#### 4. Preguntas Clave sobre los Protocolos

Respondiendo a las preguntas planteadas en el ejercicio:

- **¿Qué protocolo tradujo el nombre de dominio intranet.miempresa.local en una dirección IP?** En este escenario, la traducción fue manejada por el sistema operativo utilizando el **archivo hosts**, que simula la función del **DNS (Domain Name System)** localmente.
- **¿Qué protocolo garantizó que los datos llegaron correctamente, en orden y sin errores al servidor web?** El protocolo que garantizó la entrega fiable, ordenada y con control de errores fue **TCP (Transmission Control Protocol)**, en la capa de Transporte.
- **¿Qué protocolo se encargó de cifrar la comunicación entre tu navegador y el servidor web para proteger la privacidad de los datos?** En este escenario, dado que se accedió vía **http://** (puerto 80), **ningún protocolo se encargó de cifrar la comunicación**. Si se hubiera configurado y accedido a **https://**, el protocolo encargado sería **TLS (Transport Layer Security)**. Es importante destacar que la simulación actual es de comunicación no cifrada.
- **¿Qué protocolo es responsable del direccionamiento lógico de los paquetes para que lleguen a la IP correcta del servidor?** El protocolo responsable del direccionamiento lógico de los paquetes a la IP 127.0.0.1 es **IP (Internet Protocol)**, en la capa de Internet.

#### 5. Reflexión sobre la Importancia del Trabajo Conjunto de los Protocolos

La navegación web, incluso en un entorno simulado localmente, es un claro ejemplo de la complejidad y la interdependencia de los protocolos de red. Cada capa del modelo TCP/IP, con sus protocolos específicos, cumple una función vital que se complementa con las demás:

- **DNS** (o su simulación con el archivo hosts) traduce nombres amigables a direcciones IP, haciendo la red utilizable para los humanos.
- **IP** se encarga de que los paquetes viajen por la ruta correcta a su destino.
- **TCP** asegura que, a pesar de los desafíos de la red, la comunicación sea fiable, los datos lleguen completos y en el orden esperado.
- **HTTP** es el protocolo que permite la interacción con la aplicación web en sí. La ausencia de **TLS** en este escenario HTTP subraya la importancia de la

seguridad adicional en entornos de producción para proteger la privacidad e integridad de los datos.

Sin la colaboración de todos estos protocolos, desde la resolución de nombres hasta la entrega fiable de los datos, la experiencia de navegar por la web sería imposible. Es la sinergia de estas capas la que permite la infraestructura de Internet tal como la conocemos.